

What is claimed is

1. A memory management apparatus comprising:

a data memory which comprises a plurality of data blocks, each of which comprises a plurality of sub data blocks having a predetermined size, and
5 when there is a request for allocating memory space of a variable size, allocates memory space in units of any one of the sub data blocks and the data blocks;

a free list memory which manages an free memory space of the data memory as at least one or more lists; and

10 registers that store head location information and tail location information of the list.

2. The apparatus of claim 1, further comprising:

an address translator which performs address translation between the
15 data memory and the free list memory.

3. The apparatus of claim 1, wherein the number of entries of the free list memory is the same as the number of entries of the data memory and the entries of the free list memory and the entries of the data memory have a
20 1:1 corresponding relationship.

4. The apparatus of claim 3, wherein the start address of each entry forming the data memory is "entry number x data block size (n) + data memory start address value".

25 5. The apparatus of claim 1, wherein the data memory has a hierarchical structure, in which the data memory contains a plurality of data blocks each having memory space of n bytes when n is a power of 2 and i and j are positive integers ($i < j$), and each data block comprises a plurality of sub data
30 blocks each having a memory space of $\frac{n}{2^i}$ bytes and each sub data block

comprises a plurality of sub data blocks each having a memory space of $\frac{n}{2^j}$ bytes.

6. The apparatus of claim 1, wherein each entry forming the free
5 list memory comprises:

a plurality of bit masks each indicating whether or not a sub data block
is in use; and

a first pointer which indicates an entry located immediately after an
entry currently selected in the list.

10

7. The apparatus of claim 1, wherein each entry forming the free
list memory further comprises a second pointer indicating an entry located
immediately before an entry currently selected in the list.

8. The apparatus of claim 6, wherein according to the bit mask
15 value, the free list memory forms an n-byte entry list capable of allocating an n-
byte memory space, an $\frac{n}{2^i}$ -byte entry list capable of allocating an $\frac{n}{2^i}$ -byte
memory space, and an $\frac{n}{2^j}$ -byte entry list capable of allocating an $\frac{n}{2^j}$ -byte
memory space.

20

9. The apparatus of claim 1, wherein when the data block is
formed with X sub data blocks, the registers contain $(1 + \log_2 X)$ pointer pairs for
storing the head location information and the tail location information of the list.

25

10. The apparatus of claim 8, wherein when memory space of $\frac{n}{2^i}$
bytes is allocated, if there is no valid entry in the $\frac{n}{2^i}$ -byte entry list, the $\frac{n}{2^{i-1}}$ -
byte entry list is divided and allocated as the $\frac{n}{2^i}$ -byte memory space.

11. The apparatus of claim 8, wherein when memory space of $\frac{n}{2^j}$ bytes is allocated, if there is no valid entry in the $\frac{n}{2^j}$ -byte entry list, the $\frac{n}{2^i}$ -byte entry list is divided and allocated as the $\frac{n}{2^j}$ -byte memory space.

5

12. The apparatus of claim 8, wherein when the $\frac{n}{2^i}$ -byte memory space is deallocated, if an $\frac{n}{2^i}$ -byte memory space neighboring the deallocated memory space in the same entry is not in use, the deallocated memory space and the neighboring memory space is combined and deallocated as an $\frac{n}{2^{i-1}}$ -byte memory space.

10

13. The apparatus of claim 8, wherein when the $\frac{n}{2^j}$ -byte memory space is deallocated, if an $\frac{n}{2^j}$ -byte memory space neighboring the deallocated memory space in the same entry is not in use, the deallocated memory space and the neighboring memory space are combined and deallocated as an $\frac{n}{2^i}$ -byte memory space.

15

14. A memory allocation method comprising:

(a) when n is a power of 2 and i is a positive integer, if the size of a requested memory space for allocation is greater than $\frac{n}{2^i}$ bytes, allocating an $\frac{n}{2^{i-1}}$ -byte memory space to a valid entry existing in an $\frac{n}{2^{i-1}}$ -byte entry list managed by a free list memory; and

20

(b) if the size of a requested memory space for allocation is equal to or less than $\frac{n}{2^i}$, allocating an $\frac{n}{2^i}$ -byte memory space to a valid entry existing in an $\frac{n}{2^i}$ -byte entry list managed by the free list memory, but if there is no valid entry in the $\frac{n}{2^i}$ -byte entry list, dividing the $\frac{n}{2^{i-1}}$ -byte entry list and allocating the
5 divided $\frac{n}{2^{i-1}}$ -byte entry list as an $\frac{n}{2^i}$ -byte memory space.

15. The method of claim 14, wherein step (a) comprises:

(a-1) allocating an entry corresponding to the head location of the $\frac{n}{2^{i-1}}$ -byte entry list as the memory space, and setting a bit mask corresponding to the
10 memory space to a first value which indicates that the entry is currently in use; and

(a-2) updating the head location value of the $\frac{n}{2^{i-1}}$ -byte entry list with the location value of a next entry in the same entry list.

15 16. The method of claim 14, wherein step (b) comprises:

(b-1) determining whether or not there is a valid entry in the $\frac{n}{2^i}$ -byte entry list;

(b-2) if the result of the determination in step (b-1) indicates that there is a valid entry in the $\frac{n}{2^i}$ -byte entry list, allocating an entry corresponding to the
20 head location of the $\frac{n}{2^i}$ -byte entry list as the memory space and setting a bit mask corresponding to the memory space to the first value;

(b-3) updating the head location value of the $\frac{n}{2^i}$ -byte entry list with the location value of a next entry in the entry list to which the previous entry belongs;

(b-4) if the result of the determination in step (b-1) indicates that there is
5 no valid entry in the $\frac{n}{2^i}$ -byte entry list, allocating an entry corresponding to the head location of the $\frac{n}{2^{i-1}}$ -byte entry list as the memory space and setting the bit mask corresponding to the memory space to the first value;

(b-5) adding the location value of the allocated entry at the tail location of the $\frac{n}{2^i}$ -byte entry list; and

10 (b-6) updating the head location value of the $\frac{n}{2^{i-1}}$ -byte entry list with the location value of a next entry in the entry list to which the previous entry belongs.

17. A memory deallocation method comprising:

(a) when n is a power of 2 and i is a positive integer, if the size of a
15 deallocated memory space is greater than $\frac{n}{2^i}$ bytes, deallocating an $\frac{n}{2^{i-1}}$ -byte memory space to a data memory and including an entry corresponding to the memory space in an $\frac{n}{2^{i-1}}$ -byte entry list managed by a free list memory; and

(b) if the size of a deallocated memory space is equal to or less than $\frac{n}{2^i}$ bytes, deallocating a memory space of $\frac{n}{2^i}$ bytes to the data memory and
20 including an entry corresponding to the memory space in an $\frac{n}{2^i}$ -byte entry list managed by the free list memory, but if a neighboring memory space managed by the entry which manages the deallocated memory space is not in use, including an entry, which corresponds to a memory space obtained by

combining the deallocated memory space and the neighboring memory space,
in the $\frac{n}{2^{i-1}}$ -byte entry list.

18. The method of claim 17, wherein step (a) comprises:

5 (a-1) setting the bit mask of a data block corresponding to the deallocated memory space, to a second value indicating that the entry is not currently in use, and setting the location value of a next memory space to be used, to a third value indicating that there is no valid entry; and

(a-2) adding the deallocated memory space to the tail location of the
10 $\frac{n}{2^{i-1}}$ -byte entry list.

19. The method of claim 17, wherein step (b) comprises:

(b-1) determining whether or not $\frac{n}{2^i}$ -byte sub data blocks included in
an entry to which the deallocated memory space belongs are all in use;

15 (b-2) if the result of the determination in step (b-1) indicates that all of the sub data blocks are in use, setting the bit mask of the sub data block corresponding to the deallocated memory space to the second value, and setting the location value of a next memory space to be used to the third value;

(b-3) adding the deallocated memory space to the tail location of the
20 $\frac{n}{2^i}$ -byte entry list;

(b-4) if the result of the determination in step (b-1) indicates that not all of the sub data blocks are in use, deleting the entry corresponding to the deallocated memory space from the $\frac{n}{2^i}$ -byte entry list;

(b-5) setting the bit mask of the data block containing the deallocated
25 memory space to the second value and setting the location value of a next memory space to be used to the third value; and

(b-6) adding the deallocated memory space to the tail location of the $\frac{n}{2^{i-1}}$ -byte entry list.

20. A computer readable medium having embodied thereon a
5 computer program for the method of any one of claims 14,15,16,17,18 and 19.